

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.**

Docket No.: HI-34

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of

Se Jin HWANG

Serial No.: New U.S. Patent Application

Filed: August 30, 2001

For: CPU SCHEDULING METHOD AND APPARATUS

TRANSMITTAL OF CERTIFIED PRIORITY DOCUMENT

Assistant Commissioner of Patents  
Washington, D. C. 20231

Sir:

At the time the above application was filed, priority was claimed based on the following application:

Korean Patent Application No. 51563/2000, filed September 1, 2000.

A copy of each priority application listed above is enclosed.

Respectfully submitted,  
FLESHNER & KIM, LLP



Daniel Y.J. Kim  
Registration No. 36,186  
Laura L. Lee  
Registration No. P-48,752

P. O. Box 221200  
Chantilly, Virginia 20153-1200  
703 502-9440

Date: August 30, 2001

DYK\LLL:dep



J1046 U.S. PTO  
09/941849



# 대한민국 특허청

## KOREAN INTELLECTUAL PROPERTY OFFICE

별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto  
is a true copy from the records of the Korean Intellectual  
Property Office.

출원 번호 : 특허출원 2000년 제 51563 호  
Application Number

출원 년 월 일 : 2000년 09월 01일  
Date of Application

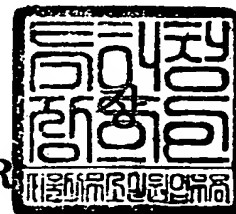
출원인 : 엘지정보통신주식회사  
Applicant(s)



2001 년 04 월 11 일

특 허 청

COMMISSIONER



【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0002
【제출일자】	2000.09.01
【국제특허분류】	H04N
【발명의 명칭】	시피유 스케줄링 방법
【발명의 영문명칭】	CPU SCHEDULING METHOD
【출원인】	
【명칭】	엘지정보통신 주식회사
【출원인코드】	1-1998-000286-1
【대리인】	
【성명】	허용록
【대리인코드】	9-1998-000616-9
【포괄위임등록번호】	1999-047173-5
【발명자】	
【성명의 국문표기】	황세진
【성명의 영문표기】	HWANG, Se Jin
【주민등록번호】	710223-1019018
【우편번호】	465-013
【주소】	경기도 하남시 덕풍3동 326-117
【국적】	KR
【취지】	특허법 제42조의 규정에 의하여 위와 같이 출원합니다. 대 리인 록 (인) 허용
【수수료】	
【기본출원료】	13 면 29,000 원
【가산출원료】	0 면 0 원
【우선권주장료】	0 건 0 원
【심사청구료】	0 항 0 원
【합계】	29,000 원
【첨부서류】	1. 요약서·명세서(도면)_1통

**【요약서】****【요약】**

본 발명은 CPU 스케줄링 방법에 관한것으로, 특히 CPU의 클럭을 프로세스의 상태 (Run, Wait, Sleep)에 따라 제어하여 CPU 전원을 효율적으로 사용함으로써 전력 소모를 최소화 하도록 한 CPU 스케줄링 방법에 관한 것이다.

스케줄러(Scheduler)함수에 시스템의 클럭 함수를 입력하는 단계와, 스케줄링이 될때까지의 대기시간(Wk)을 실시간 조건을 만족하기 위해 주어진 수행시간(Tk)과 스케줄링 오류허용치(Ek)의 오차범위의 합과 비교하는 단계와, 상기 비교한 결과에 의거 프로세스 클럭의 상태를 변환하는 단계와, 상기에 의해 변환된 스케줄링시의 클럭과 변환전의 스케줄링 클럭의 차이에 의한 흘러간 시간을 계산하여 스케줄링이 될때까지의 대기시간(Wk)을 조정하는 단계와, 새로 결정된 클럭값으로 프로세서 클럭을 설정하는 방법으로 CPU를 스케줄링 한다.

따라서, 본 발명에 의하면 시스템의 클럭함수를 이용하여 프로세스의 동작 상태를 파악하여 차별화된 전력을 각 프로세스에 공급함으로써, 프로세스의 실시간 조건을 만족 시키면서 최소의 클럭만을 유지하므로 임베디드(Embeded) 시스템등에서 전원을 장기간 사용 할 수 있다.

**【대표도】**

도 3

**【색인어】**

스케줄링방법, 전력(전원)소모

**【명세서】****【발명의 명칭】**

시피유 스케줄링 방법 {CPU SCHEDULING METHOD}

**【도면의 간단한 설명】**

도1은 종래의 CPU 동작 제어 블록도

도2는 본 발명이 수행되는 임베디드 시스템의 블록도

도3은 본 발명의 CPU 스케줄링 알고리즘

**【발명의 상세한 설명】****【발명의 목적】****【발명이 속하는 기술분야 및 그 분야의 종래기술】**

- <4> 본 발명은 CPU 스케줄링 방법에 관한것으로, 특히 CPU의 클럭을 프로세스의 상태 (Run,Wait,Idle)에 따라 제어하여 CPU 전원을 효율적으로 사용함으로써 전력 소모를 최소화 하도록 한 CPU 스케줄링 방법에 관한 것이다.
- <5> 더욱 상세하게는, 시스템의 클럭을 결정하는 지역변수(Wk,Tk,Ck,Ek)을 입력하고 그 지역변수의 값을 이용하여 프로세스 상태를 바꾸는 것이다.
- <6> 종래의 스케줄링시에는 CPU 시스템에서의 전력 소모를 위해 별도의 추가적인 하드웨어의 구성이 필요 하거나, 아니면 아예 전력 소모에 대한 것을 고려하지 않고 대용량의 전력을 모든 프로세서에 공급 하였다.
- <7> 도 1은 종래의 다중 CPU 동작 상태를 파악하는 블록도로써.
- <8> 동작을 수행하는 Qualcomm사의 CPU인 MSM3100(1,2,3)와, 상기 프로세서의 동작 상태를

과약하는 감시부(5)와, 상기 감시부의 출력에 의거 상기 CPU를 제어하는 제어부(4, PM1000)를 포함하여 구성되고 있다.

<9> 한편, 상용 운영체제인 ISI사의 pSOS와 기타 일부 실시간 운영체제 시스템은 스케줄링시에 전력소모에 대해 고려하지 않고 모두의 프로세서를 동작하고 있다.

<10> 도 1의 종래의 다중 CPU 동작 제어 상태를 설명하면 다음과 같다.

<11> 감시부(5)는 다중의 CPU(1,2,3)(MSM3100)의 동작 및 각 CPU내의 프로세서 동작상태를 감시하여 제어부(4)(PM100)에 출력한다. 제어부(4)에서는 감시부(5)에서 입력한 자료를 가지고 각 CPU(1,2,3) 및 CPU내의 동작상태를 Run, Wait, Sleep, 또는 Ready등으로 제어하여 CPU의 전력(전원) 사용을 조절하고자 하였다.

<12> 그런데 종래의 CPU의 전력소모 억제방법은 CPU의 동작 상태를 감시부(5)를 통해 입력 받아서 CPU를 제어하는 별도의 하드웨어(4)(PM100)를 설치하므로 추가적인 비용 뿐만 아니라 제조공정의 복잡화 및 부품의 소형화에 문제점이 제기 되었다.

#### 【발명이 이루고자 하는 기술적 과제】

<13> 따라서 본 발명은 CPU의 클럭을 프로세스의 상태(Run, Wait, Sleep)에 따라 각각 제어하기 위해 시스템의 클럭을 결정하는 지역변수(Wk, Tk, Ck, Ek등)를 입력하고 프로세스 상태에 따라서 지역변수 값을 이용하여 그에 의한 전력사용도 바꾸는 것이다.

#### 【발명의 구성 및 작용】

<14> 본 발명의 CPU의 스케줄링 방법은, CPU의 클럭을 프로세스의 상태에 의거 제

어하여 전원소모를 줄이기 위한 것으로, 스케줄러(Scheduler)함수에 시스템의 클럭 함수를 입력하는 단계와, 스케줄링이 될때까지의 대기시간(Wk)을 실시간 조건을 만족하기 위해 주어진 수행시간(Tk)과 스케줄링 오류허용치(Ek)의 오차범위의 합과 비교하는 단계와, 상기 비교한 결과에 의거 프로세스 클럭의 상태를 변환하는 단계와, 상기에 의해 변환된 스케줄링시의 클럭과 변환전의 스케줄링 클럭의 차이에 의한 홀러간 시간을 계산하여 스케줄링이 될때까지의 대기시간(Wk)을 조정하는 단계와, 새로 결정된 클럭값으로 프로세서 클럭을 설정하는 단계를 포함하는 것을 특징으로 하는 CPU 스케줄링 방법이다.

<15> 또한 본 발명의 CPU의 스케줄링 방법에서, 프로세서 상태는 Run, Wait, 또는 Sleep 인것을 특징으로 한다.

<16> 또한 본 발명의 CPU의 스케줄링 방법에서, 시스템의 클럭 함수는 Tk,Ck,Wk,Ek,Clock\_up(크),Clock\_advance(),Clock\_set()및 Insert\_new\_process() 인것을 특징으로 한다.

<17> 또한 본 발명의 CPU의 스케줄링 방법에서, 스케줄링이 될때까지의 대기시간(Wk)을 실시간 조건을 만족하기 위해 주어진 수행시간(Tk)과 스케줄링 오류허용치(Ek)의 오차범위의 합과 비교하여 Wk가 클때에는 Clock\_up(+1)하여 현재의 클럭보다 한단계 높은 클럭을 사용하고, Wk가 작을때에는 Clock\_up(-1)하여 현재의 클럭보다 한단계 낮은 Clock을 사용하여 프로세스를 수행하는 것을 특징으로 한다.

<18> 제2도는 본 발명이 수행되며 CPU를 내장한 임베디드(Embedded) 시스템(1)으로써, 시스템을 제어하는 CPU(2)와, 시스템을 운영하는 운영S/W(3)과, 메모리부(5)로 구성되어 있다.

- <19>      상기 CPU(2)내에는 시스템의 동작 클럭을 제어 할 수 있는 PLL 레지스터(4)가 탑재되어 실시간 제어를 위한 클럭을 CPU 내부에서 유지한다.
- <20>      상기 운영S/W(3)는 프로세스 형태의 어플리케이션(3b)과, 운영체제(3a)로 구성되며, 운영 S/W의 중핵적인 부분인 운영체제의 스케줄러는 CPU 상에서 수행되는 모든 프로세서들의 상태를 알 수 있고, 상기 결과에 따라 프로세서의 클럭을 조절한다. 운영체제내에는 프로세스 상태인 Run, Wait, Sleep, 또는 Idle등의 상태 각각에 따라 별도의 큐(Queue) 즉, Run, Wait, 또는 Sleep를 유지한다.
- <21>      본 발명의 시스템의 운영 S/W내에서 동작을 수행하는 프로세서는 다수개이며, 이로 인해 서로 다른 상태를 갖는 프로세서들의 성능을 유지하기 위해 운영체제 스케줄러 및 CPU는 클럭을 적절히 유지해야 한다. 한편 각각의 프로세서(Pj, Pk등)들은 생성시에 스케줄링의 Time Quantum을 지정 할 수 있고 이것은 스케줄러에게 스케줄링 파라미터로 인식된다.
- <22>      이하, 첨부된 도3를 참조하여 본 발명의 CPU 스케줄링 방법을 설명하면 다음과 같다.
- <23>      본 발명에서 프로세스 상태(Run, Wait, Sleep)를 변환하기 위해, 운영체제(3a)의 스케줄러 함수의 실행이 종료되기 직전에 시스템의 클럭을 결정하는 함수를 추가하여 스케줄러가 스케줄링 동안 모니터링한 결과를 가지고 프로세스 상태를 변환한다. 즉, 변환된 동작(Run) 프로세스에 주전원을 인가하고, Wait나 Sleep상태의 프로세스에는 그에 상당하는 전원을 인가하여 전력사용을 효율적으로 할 수 있다.
- <24>      본 발명의 실행을 위해 사용하는 클럭의 종류는 미리 결정되어 있어서 다양한 시

스텝 클럭을 사용할 수 있는것이 아니라 미리 결정된 클럭들 사이에서 클럭을 조절하도록 되어 있다. 예를 들면 전역변수 CLKt는 시스템내에서 사용 가능한 CPU 클럭이고, 지역변수로서 임의의 프로세스 Pk가 갖추어야 하는 클럭에 관련된 정보로는 CLKk, Tk, Ck, Wk, Ek등이 있다. 여기에서 CLKk는 프로세서 k에 해당하는 프로세서 클럭의 집합이고, Tk는 실시간 조건을 만족하기 위해서 주어진 시간이고, Ck는 현재까지 수행된 시간이고, Wk는 스케줄링이 될때까지의 대기시간이고, Ek는 스케줄링 오류 허용치 이다.

<25> 본 발명의 프로세스들의 스케줄링 방법을 구현하기 위해 사용되는 함수들에 대해 설명하면 다음과 같다.

<26> Clock\_up()함수는 해당 프로세서 클럭을 제어하기 위한 것으로, 만약 Clock\_up(+1)인 경우, 현재 Wait queue일때는 Pk가 갖는 CLKk 다음의 클럭 원소로, 만약 Clock\_up(-1)일때는 Pk가 갖는 CLKk이전의 클럭원소를 이용한다. 상기에서 다음원소를 가져올때는 클럭이 빨라지고, 이전원소를 가져올때에는 클럭이 느려지는 것이다.

<27> Clock\_advance()함수는 Run queue, Sleep queue 및 Wait queue에 존재하는 각 프로세스들의 실시간 클럭을 제어하기 위한 것으로, 모든 현재 존재하는 프로세스들을 스캔하면서 결정된 이전의 스케줄링시의 클럭과 현재의 스케줄링 클럭의 차이에 대해서 흘러간 시간을 계산하여 각각의 프로세서들의 대기시간인 Wk를 조정하는 함수이다.

<28> Clock\_set()함수는 시스템의 클럭을 변경하는 것으로, 새로 결정된 클럭인 CLKs값으로 프로세서의 클럭을 설정하는 함수이다.

<29> Insert\_new\_process()함수는 하나의 프로세스가 시스템 내부의 Run queue에 진입할시에 해당 지역변수를 초기화 하는 것으로, 새로 입력된 프로세스 Ps에 대해서 지역변수

인 CLKs를 CLK0값으로 초기화 설정하고, Ts는 사용자로부터 주어지며 Ws도 0으로 설정된다.

- <30> 도 3은 본 발명의 CPU 클럭을 조절하는 스케줄링 알고리즘을 나타낸 것이다.
- <31> 먼저, 다음과 같이 가정하고 (Pj:현재까지 수행<run>되던 프로세스), (Pk:sleep queue나 wait queue에서 하나 꺼내온 프로세스) 본 발명을 설명한다.
- <32> 첫단계(301)에서는 본 발명의 알고리즘을 운영하기 위한 시스템의 클럭 함수들 (Tk,Ck,Wk,Ek,Clock\_up(크),Clock\_advance(),Clock\_set()및 Insert\_new\_process())을 스케줄러 함수에 입력한다. 상기 함수들은 도2의 운영 S/W의 운영체제(3a)의 스케줄러 내부에 장착되어 CPU(2)상에서 수행되는 모든 프로세서들의 클럭을 제어하는데 이용된다.
- <33> 상기 단계(301)에서 입력한 함수를 이용하여, 스케줄링이 될때까지의 대기시간(Wk)이 실시간 조건을 만족하기 위해 주어진 수행시간(Tk)과 스케줄링 오류허용치(Ek)의 오차범위의 합과 비교하는 단계를 가지며, Wk가 클때에는 Clock\_up(+1)하여 현재의 클럭 상태(Pj:현재까지 수행<run>되던 프로세서) 다음의 Clock상태(Pj:Wait/Sleep 프로세서)를 수행하고, Wk가 작을때에는 Clock\_up( - 1)하여 현재의 클럭 상태(Pk:Sleep나Wait 상태의 프로세서)의 전 Clock상태(Pk:run 프로세서)를 수행하여 프로세서 동작 상태를 바꾼다. (단계302,303,304,305)
- <34> Clock\_advance()함수는 Run queue, Sleep queue 및 Wait queue에 존재하는 각 프로세서들의 실시간 클럭을 제어하기 위한 것으로, 현재 존재하는 프로세스들을 스캔하면서 결정된 이전의 스케줄링시의 클럭과 현재의 스케줄링 클럭의 차이에 대해서 흘러간 시간을 계산하여 단계 302를 수행하기 위해 각각의 프로세서들의 대기시간인 Wk를 조정한다.

다. (단계 306)

- <35> Clock\_set()함수는 시스템의 클럭을 변경하는 것으로 상기 306단계에서 조정된  $W_k$  뿐만 아니라  $CLK_k$ 의 지역변수인  $T_k, C_k, E_k$ 의 값을 새로 결정된 클럭인  $CLK_s$ 값으로 프로세서의 클럭을 설정하는 함수이다.(단계 307)
- <36> Insert\_new\_process()함수는 하나의 프로세스가 시스템 내부의 Run queue에 진입할 시에 해당 지역변수를 초기화 하는 것으로, 새로 설정된 프로세스를  $P_s$ 라 한다면 이에 대해서 시스템 클럭함수중 지역변수인  $CLK_s$ 를  $CLK_0$ 값으로 초기화 설정하고,  $T_s$ 는 사용자로부터 주어지며  $W_s$ 도 0으로 설정된다.(단계 308)
- <37> 이와 같은 과정을 통해 새로운 프로세서가 설정되고 시스템 클럭함수도 초기화 되어 본 발명의 알고리즘 과정은 종료된다.(단계 309)
- <38> 상기 과정들은 도2의 운영 S/W의 운영체제(3a)의 스케줄러 내부에 장착되어 CPU(2) 상에서 수행되며, CPU의 클럭을 시스템의 클럭함수를 이용하여 프로세스의 상태 (Run, Wait, Sleep)에 따라 각각 제어하여, 모든 프로세서에 지속적인 대전력 공급을 피하고 차별화된 전력을 공급할 수 있다.

#### 【발명의 효과】

- <39> 시스템의 클럭함수를 이용하여 프로세스의 동작 상태를 파악하여 차별화된 전력을 각 프로세스에 공급함으로써, 프로세스의 실시간 조건을 만족시키면서 최소의 클럭만을 유지하므로 임베디드(Embedded) 시스템등에서 전원을 장기간 사용 할 수 있다.

**【특허청구범위】****【청구항 1】**

CPU의 클럭을 프로세스의 상태에 의거 제어하여 전원소모를 줄이기 위한 CPU스케줄링 방법에 있어서, 스케줄러(Scheduler)함수에 시스템의 클럭 함수를 입력하는 단계와, 스케줄링이 될때까지의 대기시간( $W_k$ )을 실시간 조건을 만족하기 위해 주어진 수행시간( $T_k$ )과 스케줄링 오류허용치( $E_k$ )의 오차범위의 합과 비교하는 단계와, 상기 비교한 결과에 의거 프로세스 클럭의 상태를 변환하는 단계와, 상기에 의해 변환된 스케줄링시의 클럭과 변환전의 스케줄링 클럭의 차이에 의한 홀러간 시간을 계산하여 스케줄링이 될때까지의 대기시간( $W_k$ )을 조정하는 단계와, 새로 결정된 클럭값으로 프로세서 클럭을 설정하는 단계를 포함하는 것을 특징으로 하는 CPU 스케줄링 방법.

**【청구항 2】**

1항에 있어서, 스케줄링이 될때까지의 대기시간( $W_k$ )을 실시간 조건을 만족하기 위해 주어진 수행시간( $T_k$ )과 스케줄링 오류허용치( $E_k$ )의 오차범위의 합과 비교하여  $W_k$ 가 클때에는  $\text{Clock\_up}(+1)$ 하여 현재의 클럭 상태 다음의 Clock상태를 수행하고,  $W_k$ 가 작을 때에는  $\text{Clock\_up}(-1)$ 하여 현재의 클럭 상태 전 Clock상태를 수행하는 것을 특징으로 하는 CPU 스케줄링 방법.

**【청구항 3】**

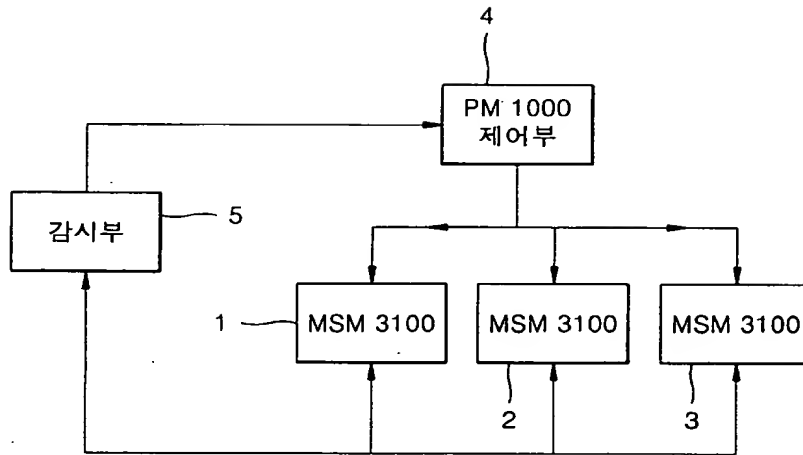
1항에 있어서, 현재의 클럭 상태가 Sleep나 wait일 경우에  $\text{Clock\_up}(+1)$ 한 상태의 클럭 상태는 Run이고,  $\text{Clock\_up}(-1)$ 한 상태의 클럭 상태는 Run 상태인 것을 특징으로 하는 CPU 스케줄링 방법.

1020000051563

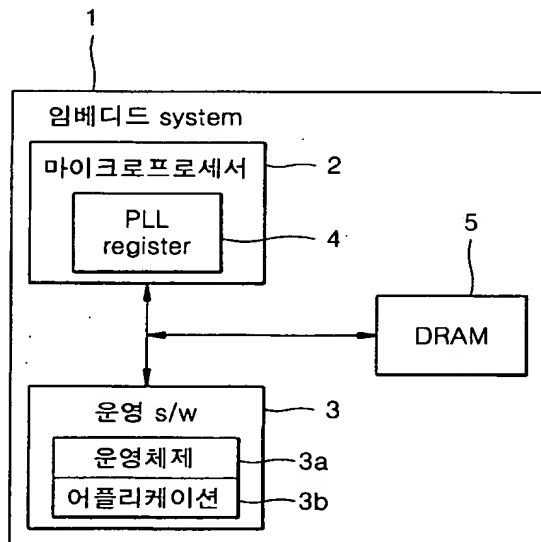
2001/4/1

## 【도면】

【도 1】



【도 2】



【도 3】

